FIG. 1

CPU — 10

HDD — 14

DSP — 16

KEYBOARD — 18

DISPLAY CONTROLLER — 20

22

PC CARD SLOT — 24

26 — PC CARD

28 PORTABLE TELEPHONE

11 BUS

12 MEMORY

APPLICATION (CPU METER) 36

OS 38

APPLICATION
· PROGRAMMING
· INTERFACE (API)
40

POWER CONTROL DEVICE DRIVER
42

POWER SOURCE UNIT 30

BATTERY 34

AC ADAPTER 32

FIG.2

- 10
- 44 CPU METER (APPLICATION・PROGRAM)
- OPERATING ENVIRONMENT DETERMINING UNIT 48
- ANIMATION SWITCHING UNIT 50
  - LIGHT LOAD PROCESSING UNIT 52
  - HEAVY LOAD PROCESSING UNIT 54
- INITIALIZING UNIT 46
- API 40
- POWER CONTROL DEVICE DRIVER 42
- FILE DRIVER 56
- DISPLAY DRIVER 58
- OS 38
- DISPLAY 22
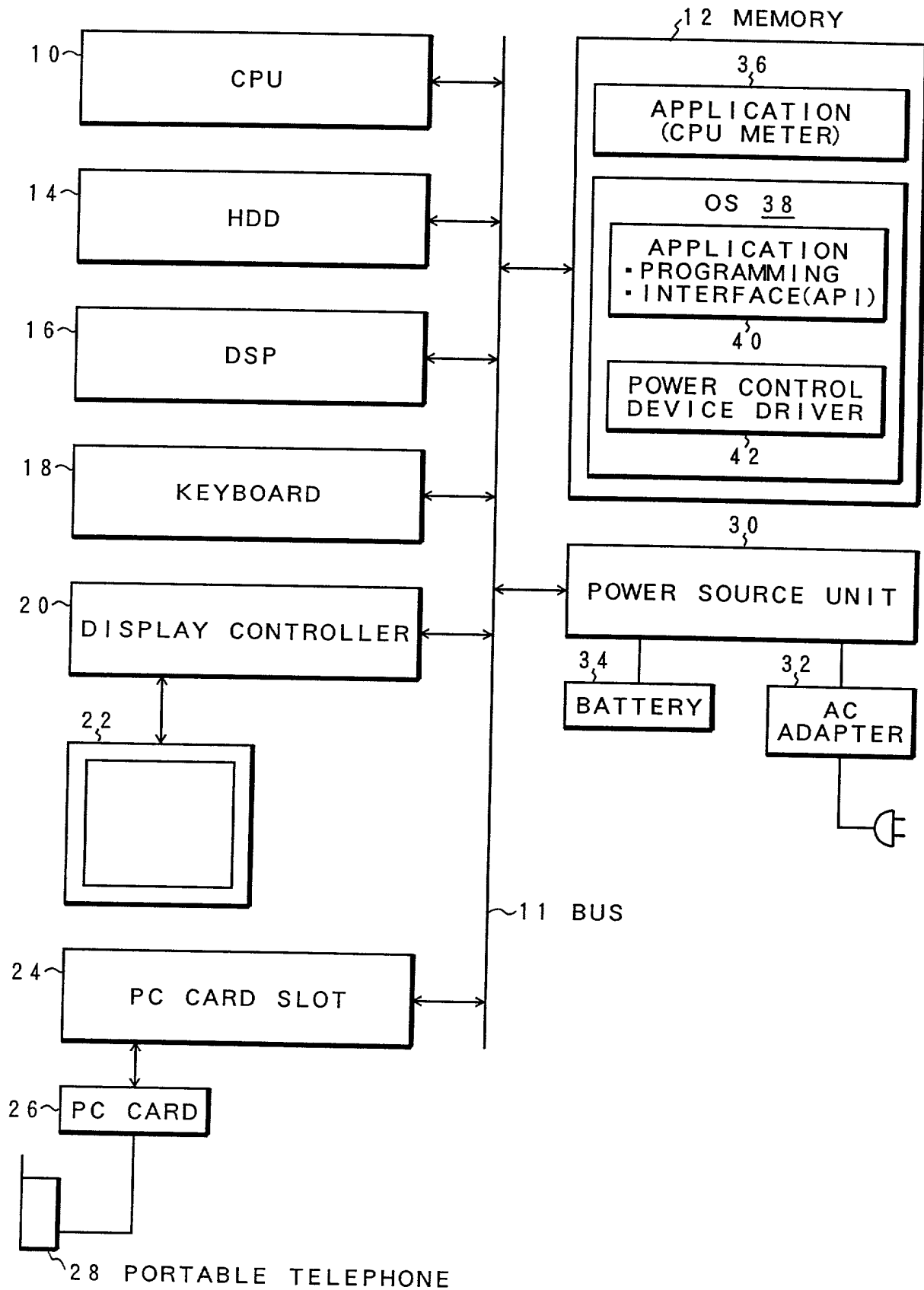- IMAGE FILE 60
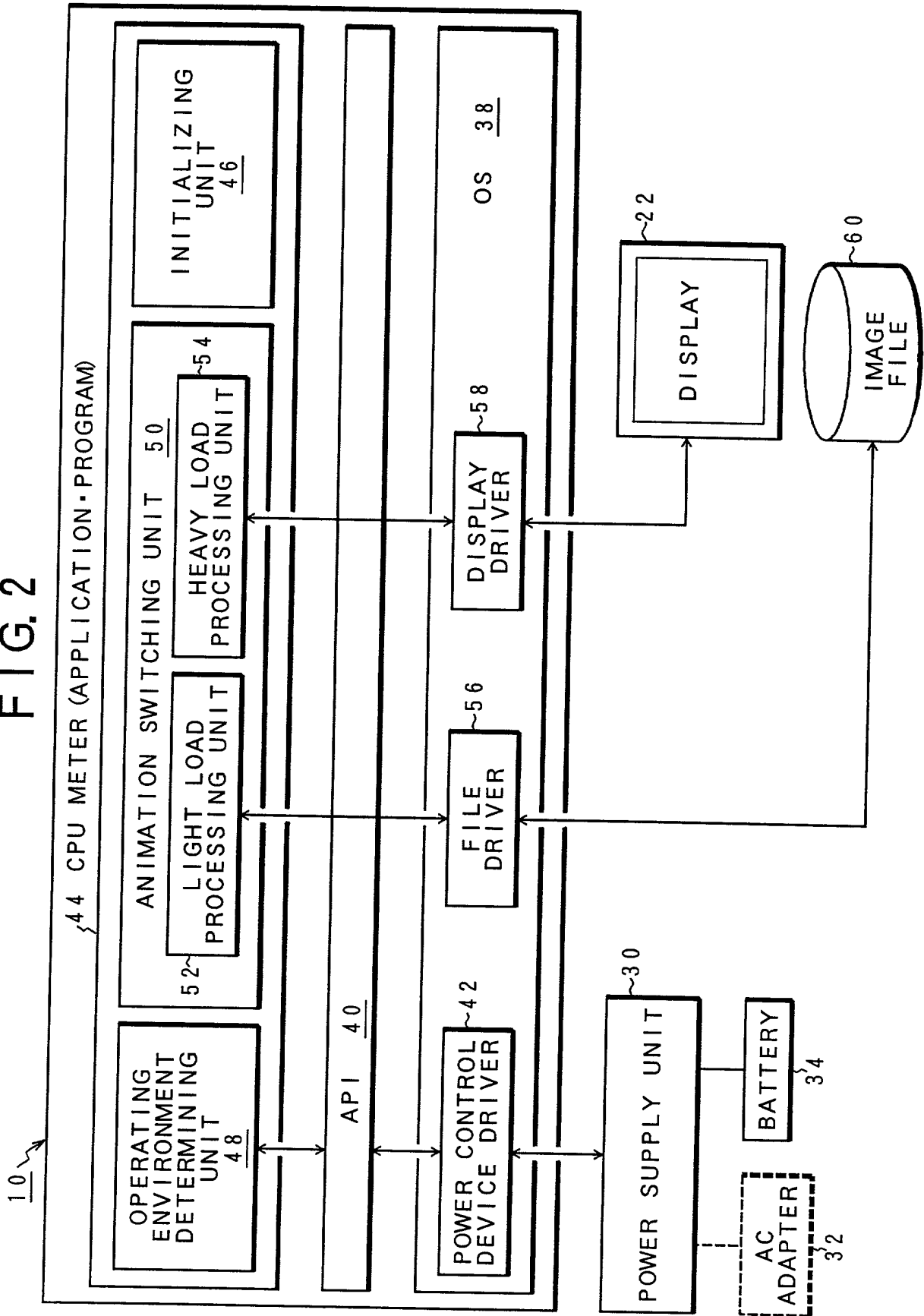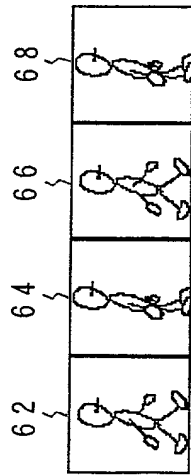- POWER SUPPLY UNIT 30
- BATTERY 34
- AC ADAPTER 32

F I G. 3A
BASIC IMAGES

F I G. 3B
ANIMATION

F I G. 3C
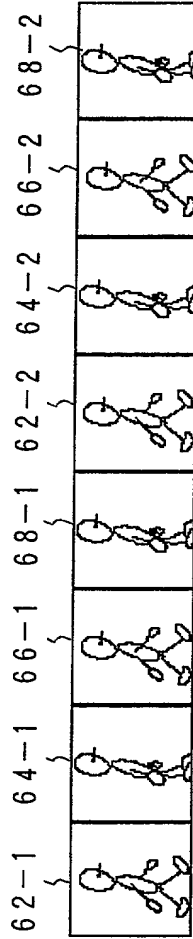STILL IMAGE

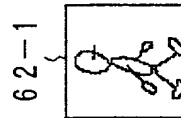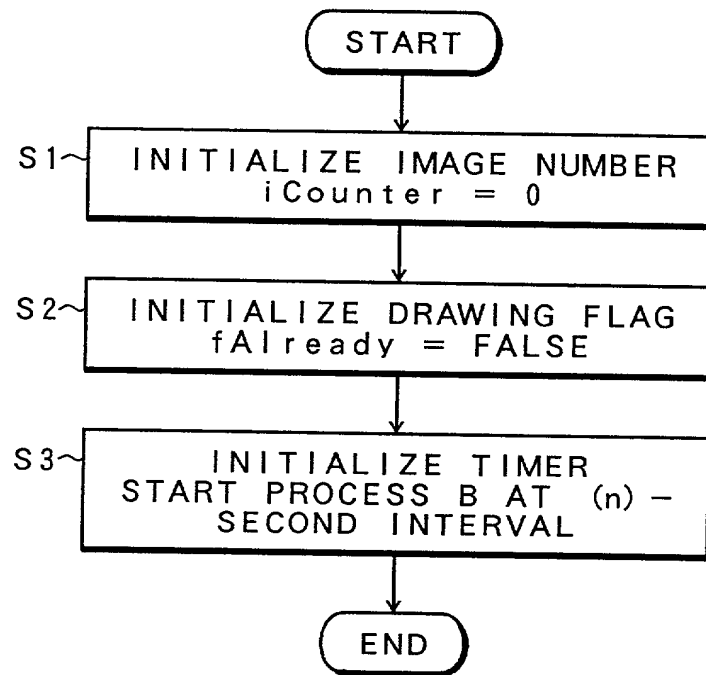# F I G. 4

```
                    ┌─────────────┐
                    │    START    │
                    └─────────────┘
                           │
                           ▼
        ┌──────────────────────────────────────┐
S1~     │   INITIALIZE  IMAGE  NUMBER          │
        │         iCounter  =  0               │
        └──────────────────────────────────────┘
                           │
                           ▼
        ┌──────────────────────────────────────┐
S2~     │   INITIALIZE  DRAWING  FLAG          │
        │         fAlready  =  FALSE           │
        └──────────────────────────────────────┘
                           │
                           ▼
        ┌──────────────────────────────────────┐
S3~     │         INITIALIZE  TIMER            │
        │   START  PROCESS  B  AT   (n) −      │
        │          SECOND  INTERVAL            │
        └──────────────────────────────────────┘
                           │
                           ▼
                    ┌─────────────┐
                    │     END     │
                    └─────────────┘
```

F I G. 5

START

S1 — OBTAIN SYSTEM POWER ENVIRONMENT
::GetSystemPowerStatus (&sps)

S2 — AC BEING CONNECTED ?

YES → DISPLAY IMAGE NUMBER iCounter — S6
ShowImage (hBMP [iCounter])

INCREMENT IMAGE NUMBER — S7
++iCounter

S8 — IMAGE NUMBER EXCEED MAXIMUM VALUE?

NO

YES → RE-INITIALIZE IMAGE NUMBER — S9
iCounter = 0

NO

S3 — fAlready = TRUE ?

YES

NO → DISPLAY IMAGE NUMBER 0 — S4
ShowImage (hBMP [0])

S5 — SET DRAWING FLAG
fAlready = TRUE

END

FIG. 6A

List. 1

```
void sample()
{
//---------------------------------------------
// #a  SYSTEM POWER STATUS STORAGE AREA
//---------------------------------------------
SYSTEM-POWER-STATUS sps;
//---------------------------------------------
// #b  OBTAIN SYSTEM POWER STATUS
//---------------------------------------------
::GetSystemPowerStatus( &sps );
//---------------------------------------------
// #c  DRAWING FLAG WHICH IS SET(INITIALIZED ONLY ONCE)
//---------------------------------------------
static BOOL  fAlready = FALSE;
//---------------------------------------------
// #d  SYSTEM BEING OPERATED ON BATTERY?
//---------------------------------------------
if ( AC-LINE-OFFLINE! = sps.ACLineStatus ) {
        //---------------------------------------------
        // #e1  NO IMAGE DRAWN
        //---------------------------------------------
        if ( !fAlready ) {
                //---------------------------------------------
                // #e2  DRAW IMAGE
                //---------------------------------------------
                ShowImage( hBMP[0] );
                //---------------------------------------------
                // #e3  ALREADY DRAWN
                //---------------------------------------------
        } else {
```

FIG. 6B
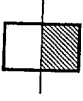
```
        //----------------------------------
        // #e4  NO OPERATION
        //----------------------------------
    }
    //----------------------------------------------------
    // #f1  SYSTEM BEING OPERATED ON AC ADAPTER
    //----------------------------------------------------
    } else {
        //----------------------------------------------------
        // #f2  ANIMATION COUNTER(INITIALIZED ONLY ONCE)
        //----------------------------------------------------
        static int iCounter = 0;
        //----------------------------------
        // #f3  DRAW IMAGE
        //----------------------------------
        ShowImage( hBMP[i] );
        //----------------------------------
        // #f4  INCREMENT ANIMATION COUNTER
        //----------------------------------
        if ( ANIMATION_MAX < ++iCounter ) {
            //----------------------------------------------------------------
            // #f5  IF LAST IMAGE HAS BEEN DRAWN, RETURN TO THE FIRST
            //----------------------------------------------------------------
            iCounter = 0;
        }
    }
    //----------------------------------
    // #g  ALREADY DRAWN
    //----------------------------------
    fAlready = TRUE;

    return;
```

FIG. 7A

74-1

70 APPLICATION WINDOW

Fujitsu CPU Meter

CPU METER
DISPLAY AREA 72

533MHz   100%

FIG. 7B

74-2

70

Fujitsu CPU Meter

72

533MHz   100%

FIG. 7C

74-3

70

Fujitsu CPU Meter

72

533MHz   100%

FIG. 7D

74-4

70

Fujitsu CPU Meter

72

533MHz   100%

FIG. 7E

74-5

70

Fujitsu CPU Meter

72

533MHz   100%

F I G. 8

74－5

70

Fujitsu CPU Meter

72 ─ 533MHz   100%
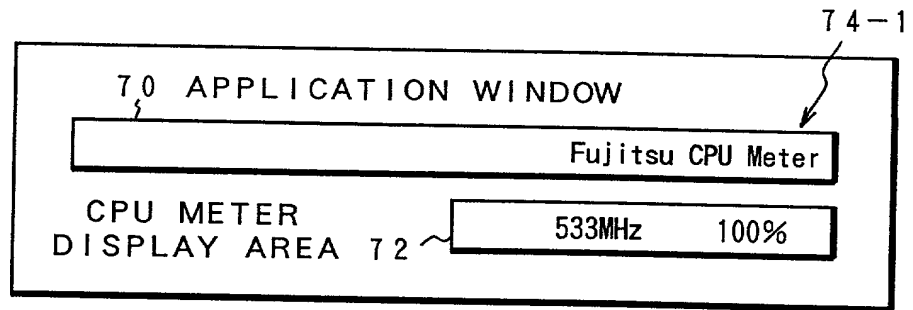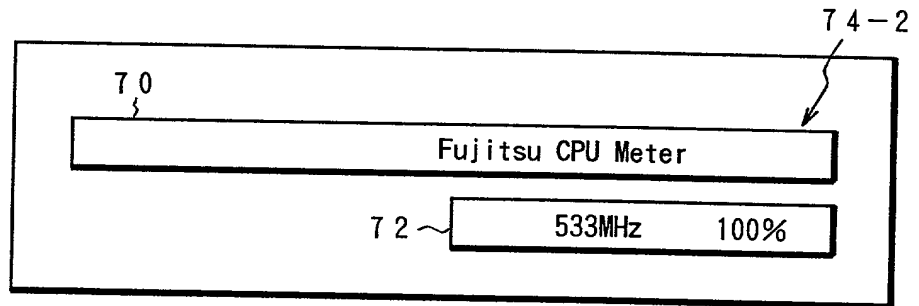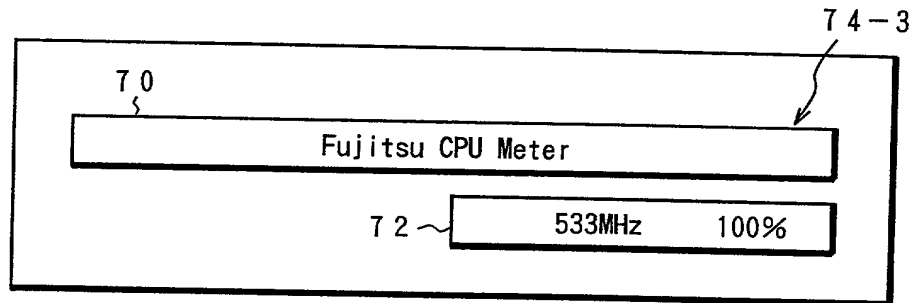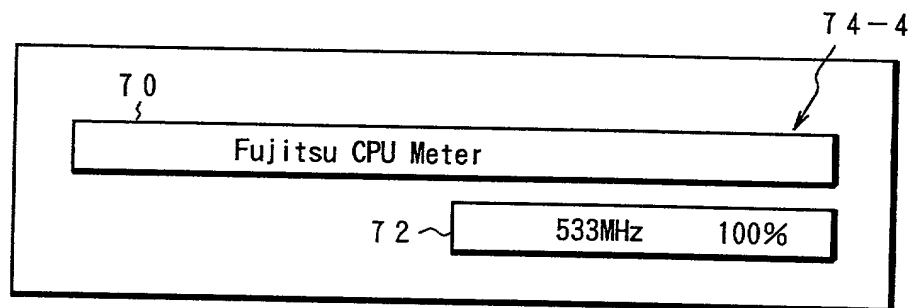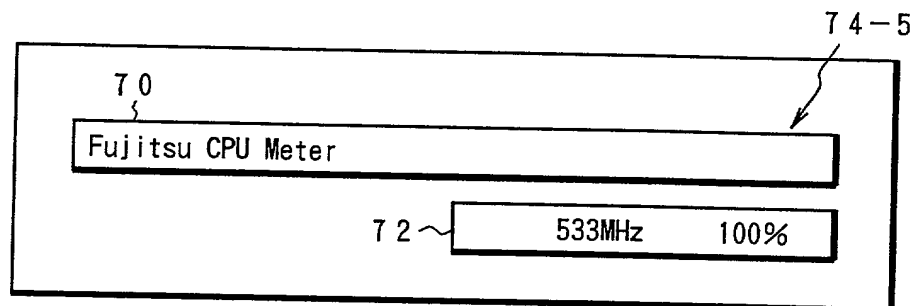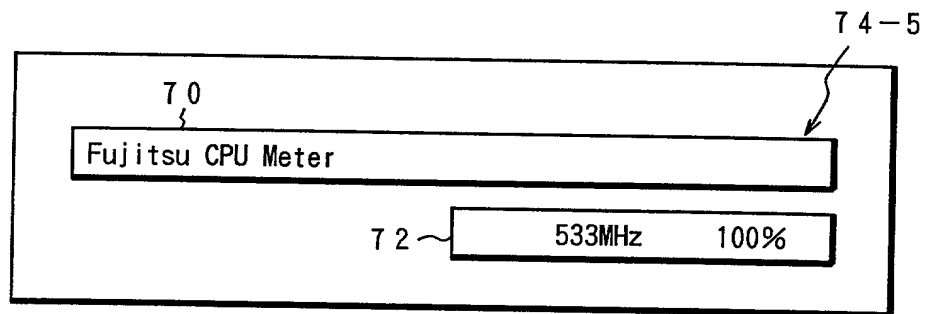
FIG. 9A

FIG. 9B

FIG. 9C

FIG. 10

F I G. 1 1

```
         ( START : CALL LIBRARY )
                    │
                    ▼
       ┌──────────────────────────────┐
S1~    │   INITIALIZE  IMAGE  NUMBER   │
       │        iCounter = 0           │
       └──────────────────────────────┘
                    │
                    ▼
       ┌──────────────────────────────┐
S2~    │   INITIALIZE  DRAWING  FLAG   │
       │       fAlready = FALSE        │
       └──────────────────────────────┘
                    │
                    ▼
       ┌──────────────────────────────┐
S3~    │       INITIALIZE  TIMER       │
       │  START PROCESS B AT uiInterval│
       └──────────────────────────────┘
                    │
                    ▼
       ┌──────────────────────────────┐
S4~    │ INITIALIZE MAXIMUM VALUE OF   │
       │        IMAGE  NUMBER          │
       │  MAXIMUM VALUE = iNumImage    │
       └──────────────────────────────┘
                    │
                    ▼
       ┌──────────────────────────────┐
S5~    │         SET hTarget           │
       │ TO WINDOW AS DRAWING TARGET   │
       └──────────────────────────────┘
                    │
                    ▼
         ( END : RETURN TO LIBRARY )
```

FIG. 12

```
        ( START )
            │
            ▼
S1~ ┌─────────────────────────┐
    │      START DIALOG X      │
    └─────────────────────────┘
            │
            ▼
S2~ ┌─────────────────────────┐
    │  STORE DIALOG SELECTION  │
    │   RESULT INTO REGISTRY Y │
    └─────────────────────────┘
            │
            ▼
S3~ ┌─────────────────────────┐
    │      ENDING PROCESS      │
    └─────────────────────────┘
            │
            ▼
         ( END )
```

FIG. 13

78

┌──┐
│ PROCESS DYNAMIC SWITCHING PROPERTY   │
├──┤
│   │
│ TO ALLOW SWITCH BETWEEN PROCESS OF HEAVY LOAD   │
│ ON CPU AND PROCESS OF LIGHT LOAD ON CPU   │
│ ACCORDING TO ENVIRONMENT, CHECK THE CHECKBOX.   │
│ IN THIS CASE, ANIMATION MAY STOP.   │
│   │
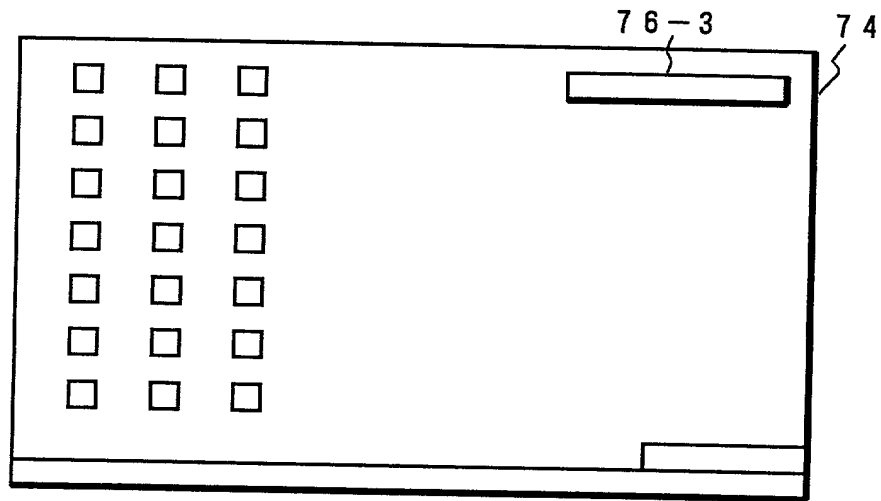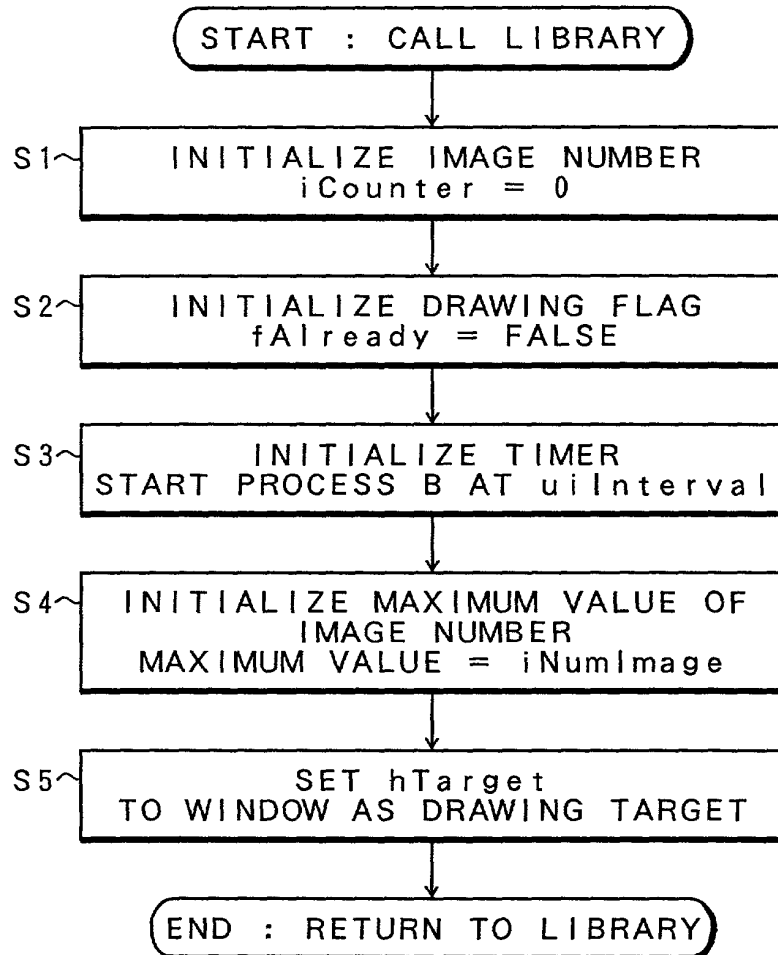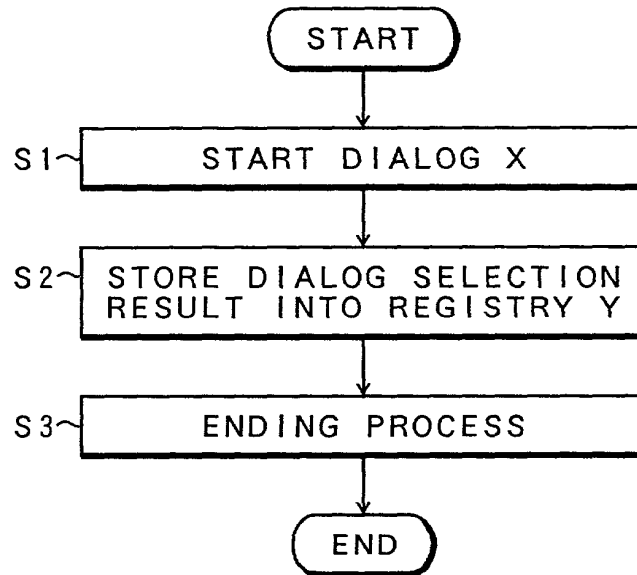│   │
│   ☐   ALLOW SWITCH BETWEEN PROCESS OF HEAVY   │
│   80   LOAD ON CPU AND PROCESS OF LIGHT LOAD   │
│   ON CPU ACCORDING TO ENVIRONMENT.   │
│   │
│   │
│   82~ ┌─────────┐   │
│   │ CLOSE │   │
│   └─────────┘   │
└──┘

FIG.14

START

S0 — REGISTRY Y = 1 ?
- NO → S6
- YES ↓

S1 — OBTAIN SYSTEM POWER ENVIRONMENT ::GetSystemPowerStatus (&sps)

S2 — AC BEING CONNECTED ?
- YES → S6
- NO ↓

S3 — fAlready = TRUE ?
- YES → S5
- NO ↓

S4 — DISPLAY IMAGE NUMBER 0 ShowImage (hBMP [0])

S5 — SET DRAWING FLAG fAlready = TRUE

END

S6 — DISPLAY IMAGE NUMBER iCounter ShowImage (hBMP [iCounter])

S7 — INCREMENT IMAGE NUMBER ++iCounter

S8 — IMAGE NUMBER EXCEED MAXIMUM VALUE?
- YES → S9
- NO → S5

S9 — RE-INITIALIZE IMAGE NUMBER iCounter = 0